

## CLAIMS

What is claimed is:

- 1 1. An operational data store, comprising:  
2 an insert table for storing new data;  
3 a history table for storing historical data; and  
4 transfer logic for periodically transferring new data from the insert table  
5 to the history table.
- 1 2. The operational data store of Claim 1, wherein the history table is partitioned.
- 1 3. The operational data store of Claim 2, wherein the history table is partitioned by  
2 range.
- 1 4. The operational data store of Claim 2, wherein each partition is further sub-  
2 partitioned.
- 1 5. The operational data store of Claim 4, wherein each partition is sub-partitioned  
2 into a number of sub-partitions equal to the number of database server instances.
- 1 6. The operational data store of Claim 5, wherein each sub-partition of a partition is  
2 associated with a database server instance.
- 1 7. The operational data store of Claim 4, wherein each sub-partition of a partition is  
2 associated with a database server instance.
- 1 8. The operational data store of Claim 1, the transfer logic comprising:  
2 a secondary table;

3 fill logic for filling the secondary table with selected data from the insert  
4 table; and  
5 secondary transfer logic for transferring the secondary table into the  
6 history table, the selected data thereby being transferred into the history table.

1 9. The operational data store of Claim 8, wherein the history table has an indexing  
2 scheme, the secondary transfer logic further comprising:  
3 indexing logic for applying the history table indexing scheme to the  
4 secondary table.

1 10. The operational data store of Claim 9, wherein the indexing logic applies the  
2 history table indexing scheme to the secondary table prior to transferring the  
3 secondary table into the history table.

1 11. The operational data store of Claim 8, the secondary transfer logic further  
2 comprising:  
3 table logic for creating a new partition the history table, the new partition  
4 for swapping with the secondary table.

1 12. The operational data store of Claim 11, wherein the secondary transfer logic  
2 swaps the secondary table and the new partition by exchanging respective  
3 pointers.

1 13. The operational data store of Claim 1, further comprising:  
2 a query engine for applying a database query to both the history table and  
3 the insert table.

1 14. The operational data store of Claim 1, further comprising:  
2 an aggregation buffer for accumulating new data; and

3 an aggregator for batching the accumulated data and transferring the  
4 batched data into the insert table with a single database access.

1 15. The operational data store of Claim 14, wherein the aggregator transfers a batch  
2 of new data into the insert table when the batch surpasses a maximum size.

1 16. The operational data store of Claim 15, wherein batch size is measured  
2 according to a number data bytes.

1 17. The operational data store of Claim 15, wherein batch size is measured  
2 according to a number of records.

1 18. The operational data store of Claim 15, wherein the maximum size is  
2 configurable.

1 19. The operational data store of Claim 14, wherein the aggregator transfers batches  
2 of new data into the insert table at regular intervals, defined by a given period.

1 20. The operational data store of Claim 19, wherein the period is configurable.

1 21. The operational data store of Claim 14, wherein the aggregator transfers batches  
2 of new data into the insert table when the aggregation buffer surpasses a given  
3 maximum buffer size.

1 22. The operational data store of Claim 21, wherein the maximum buffer size is  
2 configurable.

1 23. The operational data store of Claim 1, further comprising:

2 a throttler for throttling transactions of different classes independently to  
3 achieve a desired level of service.

1 24. The operational data store of Claim 23, wherein a first transaction class is a  
2 query, and a second transaction class is an insert, the throttler throttling queries  
3 so that inserts can be executed at at least the desired level of service.

1 25. The operational data store of Claim 23, wherein at least one transaction class  
2 comprises plural transaction types, the throttler throttling transactions of  
3 different types independently.

1 26. The operational data store of Claim 1, further comprising:  
2 a plurality of processor nodes configured as a processor cluster, wherein  
3 distinct database server instances are associated with distinct processor nodes of  
4 the processor cluster.

1 27. The operational data store of Claim 26, wherein the history table is partitioned,  
2 each partition is further sub-partitioned into a number of sub-partitions, and the  
3 number of sub-partitions is responsive to the number of database server  
4 instances.

1 28. The operational data store of Claim 27, wherein the number of sub-partitions of  
2 each partition is equal to the number of database server instances.

1 29. The operational data store of Claim 1, wherein data from the insert table is  
2 transferred to the history table at regular intervals.

1 30. The operational data store of Claim 29, wherein the intervals are configurable.

1 31. The operational data store of Claim 29, wherein the intervals are different for  
2 different tables.

1 32. A method for maintaining an operational data store, comprising:  
2 inserting new data into an insert table;  
3 periodically transferring data from the insert table to a history table.

1 33. The method of Claim 32 further comprising:  
2 partitioning the history table.

1 34. The method of Claim 33, wherein the history table is partitioned according to  
2 range.

1 35. The method of Claim 33, further comprising:  
2 sub-partitioning each partition.

1 36. The method of Claim 33, wherein each partition is sub-partitioned into a number  
2 of sub-partitions equal to the number of database server instances.

1 37. The method of Claim 36, further comprising:  
2 associating each sub-partition of a partition with a database server  
3 instance.

1 38. The method of Claim 35, further comprising:  
2 associating each sub-partition of a partition with a database server  
3 instance.

1 39. The method of Claim 32, further comprising:  
2 creating a secondary table;

3           filling the secondary table with selected data from the insert table; and  
4           transferring the secondary table into the history table, the selected data  
5           thereby being transferred into the history table.

1   40.   The method of Claim 39, wherein the history table has an indexing scheme, the  
2           method further comprising:  
3           applying the history table indexing scheme to the secondary table.

1   41.   The method of Claim 40, wherein the history table indexing scheme is applied to  
2           the secondary table prior to transferring the secondary table into the history  
3           table.

1   42.   The method of Claim 39, further comprising:  
2           creating a new partition in the history table, wherein the secondary table  
3           is transferred by being swapped with the new partition.

1   43.   The method of Claim 42, wherein the secondary table and new partition are  
2           swapped by exchanging respective pointers.

1   44.   The method of Claim 32, further comprising:  
2           applying a database query to both the history table and the insert table.

1   45.   The method of Claim 32, further comprising:  
2           aggregating new data into batches; and  
3           inserting the batched new data into the insert table with a single database  
4           access.

1   46.   The method of Claim 45, wherein a batch of new data is transferred into the  
2           insert table when the batch surpasses a maximum size.

- 1 47. The method of Claim 46, wherein batch size is measured according to a number  
2 data bytes.
- 1 48. The method of Claim 46, wherein batch size is measured according to a number  
2 of records.
- 1 49. The method of Claim 46, wherein the maximum size is configurable.
- 1 50. The method of Claim 45, wherein batches of new data are transferred into the  
2 insert table at regular intervals, defined by a given period.
- 1 51. The method of Claim 50, wherein the period is configurable.
- 1 52. The method of Claim 45, further comprising:  
2 aggregating the batches of new data in an aggregation buffer, wherein the  
3 batches are transferred into the insert table when the aggregation buffer  
4 surpasses a given maximum buffer size.
- 1 53. The method of Claim 52, wherein the maximum buffer size is configurable.
- 1 54. The method of Claim 32, further comprising:  
2 throttling transactions of different classes independently to achieve a  
3 desired level of service.
- 1 55. The method of Claim 54, wherein a first transaction class is a query, and a  
2 second transaction class is an insert, and queries are throttled so that new data  
3 can be inserted at at least the desired level of service.

1 56. The method of Claim 54, wherein at least one transaction class comprises plural  
2 transaction types which are independently throttled.

1 57. The method of Claim 32, further comprising:  
2 configuring plural processor nodes as a processor cluster; and  
3 executing distinct database server instances on distinct processor nodes  
4 of the processor cluster.

1 58. The method of Claim 57, further comprising:  
2 partitioning the history table; and  
3 sub-partitioning each partition into a number of sub-partitions, wherein  
4 the number of sub-partitions is responsive to the number of database server  
5 instances.

1 59. The method of Claim 58, the number of sub-partitions of each partition is equal  
2 to the number of database server instances.

1 60. A method for operating an operational data store, comprising:  
2 creating a new partition in a composite-partitioned history table;  
3 creating a partitioned temporary table;  
4 filling the temporary table with data from an insert table;  
5 exchanging the temporary table with the new partition; and  
6 receiving a query and applying the query to both the history table and the  
7 insert table.

1 61. The method of claim 60, further comprising:  
2 creating a new partition in the insert table based on values from an  
3 existing partition; and  
4 dropping the existing partition.



1 62. An operational data store, comprising:  
2 means for inserting new data into an insert table;  
3 means for periodically transferring data from the insert table to a history  
4 table; and  
5 means for applying a database query to both the history table and the  
6 insert table.

1 63. The operational data store of Claim 62, further comprising:  
2 means for batching new data; and  
3 means for inserting the batched new data into the insert table with a  
4 single database access.

1 64. The operational data store of Claim 62, further comprising:  
2 means for throttling transactions of different types independently to  
3 achieve a desired level of service.

1 65. An operational data store, comprising:  
2 means for creating a new partition in a composite-partitioned history  
3 table;  
4 means for creating a partitioned temporary table;  
5 means for filling the temporary table with data from an insert table;  
6 means for exchanging the temporary table with the new partition; and  
7 means for receiving a database query and applying said query to both the  
8 history table and the insert table.

1 66. A computer program product for operating an operational data store, the  
2 computer program product comprising a computer usable medium having  
3 computer readable code thereon, including program code which:

4 inserts new data into an insert table;  
5 periodically transfers data from the insert table to a history table; and  
6 applies a database query to both the history table and the insert table.

1 67. The computer program product of Claim 66, wherein the program code further:  
2 batches new data; and  
3 inserts the batched new data into the insert table with a single database  
4 access.

1 68. The computer program product of Claim 66, wherein the program code further:  
2 throttles transactions of different types independently to achieve a  
3 desired level of service.

1 69. A computer program product for operating an operational data store, the  
2 computer program product comprising a computer usable medium having  
3 computer readable code thereon, including program code which:  
4 creates a new partition in a composite-partitioned history table;  
5 creates a partitioned temporary table;  
6 fills the temporary table with data from an insert table;  
7 exchanges the temporary table with the new partition; and  
8 receives queries and applies said queries to both the history table and the  
9 insert table.

1 70. A system for producing a desired level of service in a mixed workload  
2 environment, comprising:  
3 a high-speed insert operational data store (ODS);  
4 a throttler for throttling selected transactions to the ODS; and  
5 an aggregator for accumulating transactions into batches and inserting  
6 each of the batches into the ODS using a single database transaction per batch.

1 71. The system of Claim 70, wherein the mixed workload environment includes at  
2 least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup  
3 processes and extract/translate/load transactions.

1 72. A method for producing a desired level of service in a mixed workload  
2 environment, comprising:  
3 insert transactions into an operational data store (ODS) at a high-speed;  
4 throttling selected transactions to the ODS;  
5 accumulating transactions into batches; and  
6 inserting each of the batches into the ODS using a single database  
7 transaction per batch.

1 73. The method of Claim 72, wherein the mixed workload environment includes at  
2 least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup  
3 processes and extract/translate/load transactions.